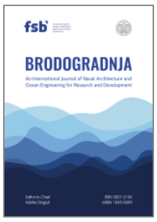
University of Zagreb
Faculty of Mechanical
Engineering and Naval
Architecture

journal homepage: www.brodogradnja.fsb.hr

Brodogradnja

An International Journal of Naval Architecture and
Ocean Engineering for Research and Development



Reinforcement learning-driven continuous maneuvering decision system for maritime collision prevention using proximal deterministic policy gradient

Xiao Yang[1], *Chunlei Wang*[1,*], *Lei Zhou*[2], *Haiyan Wang*[2], *Fengying Wang*[2]

[1] *School of Information and Engineering, Suqian University, Huanghe Road, 23800, Suqian City, Jiangsu Province, China*
[2] *Jiangsu Province Engineering Research Center of Smart Poultry Farming and Intelligent Equipment, Suqian University, Huanghe Road, 23800, Suqian City, Jiangsu Province, China*

ARTICLE INFO

Keywords:
Continuous ship steering control
Deep reinforcement learning
Pathwise Derivative Policy Gradient
MMG model
Policy network

ABSTRACT

Continuous ship steering control is a highly nonlinear and complex task, as it is subject to wave and wind disturbances. It is also crucial for timely obstacle avoidance and effective vessel maneuvering. Reinforcement learning (RL) combined with deep neural networks (DNNs) has demonstrated significant potential in controlling systems with nonlinear dynamics, making it well-suited for decision-making and planning in such complex scenarios. However, existing research struggles to ensure optimal control performance. To address this limitation, this paper proposes an improved deep reinforcement learning approach based on the Pathwise Derivative Policy Gradient (PDPG) algorithm to enable intelligent collision avoidance for continuous ship steering. The proposed method leverages the MMG model as the foundation for learning a steering control strategy using DNNs, comprehensively considers various control actions, and evaluates steering performance through a dedicated evaluation network. To enhance the policy network's representational capacity and balance exploration and exploitation, the PDPG algorithm's policy network structure is optimized. Additionally, an adaptive exploration rate and a dynamic balancing algorithm for random strategies are introduced to fine-tune the exploration-exploitation trade-off. The improved method's performance is verified through simulations of continuous ship steering control.

1. Introduction

The terrain near island capes is often complex, with geographical features forming curved waterways, such as the Luotou Waterway in Zhoushan. These bends present challenging flow conditions, making vessels prone to capsizing or running aground and thus posing significant navigation risks. Curved waterways can be single, continuous, or forked, all of which require precise and continuous ship steering. Compared to straight waterways, curved ones are generally narrower, restricting the manoeuvring space available to vessels. Ships must ensure safe passage through these narrow channels to avoid collisions with other vessels or the shoreline.

Beyond these challenges, curved waterways are also affected by meteorological conditions, which can cause ships to drift or deviate from their intended course. This requires timely adjustments to both speed and

[*] Corresponding author.
E-mail address: chunleiwang2022@163.com

https://doi.org/10.21278/brod77316
Received 30 October 2025; Received in revised form 25 February 2026; Accepted 6 March 2026
Available online 20 March 2026
ISSN 0007-215X; eISSN 1845-5859

heading. Therefore, when encountering sudden dynamic obstacles in curved waterways, ships must make rapid, stable steering and collision-avoidance decisions to prevent accidents.

Ship steering control is crucial for ensuring both safe and economical navigation. However, due to factors such as large inertia, nonlinearity, slow time-varying model parameters, and disturbances in ship motion, designing an effective controller remains challenging. Typically, the proportional, integral, and derivative (PID) parameters of the controller are preset, and professional operators must estimate and adjust them using specific mathematical formulas. The ship's navigation environment is inherently uncertain, and the hydrodynamic changes affecting the vessel are complex. As a result, PID control systems have notable limitations, leading to deficiencies in ship heading control systems [1, 2].

To enhance performance and robustness under complex disturbances, several nonlinear control algorithms have been implemented to address these challenges [3, 4]. For instance, Zhang et al. [5] proposed an Active Disturbance Rejection Control (ADRC) algorithm based on nonlinear feedback to tackle issues such as external disturbances, internal model uncertainties, and excessive rudder angle input during ship heading maintenance. This ADRC algorithm employs a linear state observer to estimate external disturbances and internal uncertainties, allowing controller adjustments that minimize excessive rudder angle inputs. Guan et al. [6] highlighted the importance of state observers in estimating ship motion-related parameters, particularly the Nomoto index of ship manoeuvrability, which significantly influences the performance of the designed steering controller. However, due to model errors and external environmental interference during navigation, this index cannot be directly observed. To address this, they developed an adaptive, robust ship steering controller that uses closed-loop gain shaping and an extended Kalman filter for online identification.

Zhang et al. further studied the switching control of a ship course-keeping autopilot (SCKA) in the event of steering gear bias failure. Such failures can cause the vessel to deviate from its intended course, especially during course maintenance. Consequently, they proposed an SCKA switching control method that incorporates steering gear bias failure detection and fault alarms, along with an augmented fault observer (AFO) to estimate unknown steering gear bias failures and make necessary corrections when they occur [7]. While existing linear and nonlinear control methods have demonstrated considerable performance in ship heading control, their effectiveness is often constrained by fixed structures and parameters.

Controller parameter optimization is a prominent research area, with various optimization algorithms employed to enhance the robustness and performance of control methods [8, 9]. However, controllers relying on fixed optimization parameters may not achieve optimal performance, as different control gains are required under varying conditions. Inspired by advances in artificial intelligence, reinforcement learning algorithms have been integrated with control theory to develop innovative control strategies that maintain optimal performance in real time. Chen et al. [10] proposed an online parameter adjustment method for linear active disturbance rejection control (LADRC) based on Q-learning, applying it to ship heading control with a simplified response model. Nonetheless, the Q-learning algorithm requires manual partitioning of the controlled object's state and specification of discrete actions, specifically controller parameters. This poses a limitation: as the number of divided states and specified actions increases, the algorithm's complexity grows significantly. Additionally, because the controller operates using predetermined discrete actions, its parameters can only assume specific values, introducing subjectivity. Sawada et al. [11] introduced an automatic ship collision avoidance algorithm utilizing deep reinforcement learning (DRL) within a continuous action space. Their approach, termed Obstacle Zone by Target (OZT), calculates the potential collision area based on the ship's dynamic information. Meanwhile, the DRL agent employs a virtual grid sensor to detect the approach of multiple vessels. Gökşu et al. [12] utilized fuzzy Bayesian networks to assess risks and investigate the root causes of ship steering gear failures, constructing a Bayesian network using NETICA software to analyze the impact of these root causes. Zhang et al. [13] explored the fault estimation problem and the fuzzy active fault-tolerant control design for ship steering autopilot systems, which are characterized by difficult-to-measure states, actuator (rudder), and sensor (compass) faults. However, these studies do not account for changes in the continuous action space resulting from variations in ship speed and steering.

The interactive learning between deep reinforcement learning and the environment effectively addresses the dynamically changing navigation requirements of curved waterways. By analyzing the motion state of ship steering control through steering control and collision avoidance simulations, encompassing the ship's position, speed, and rudder angle, the performance of the deep reinforcement learning algorithm can be significantly enhanced. The state and action spaces define the critical information required for continuous steering collision avoidance: the state space includes the ship's current state, while the action space comprises the available manoeuvres for steering the vessel. A reward function is established to evaluate the effectiveness of steering and collision-avoidance decisions. Using the current environmental state, along with the state, action, and reward spaces, the deep reinforcement learning algorithm learns to make intelligent, continuous, steering-based collision-avoidance decisions. This paper adapts these mechanisms to task-driven continuous ship steering in curved waterways, incorporating joint rudder–speed control and risk-driven encounters. A validated integration demonstrates that this specific combination is necessary to achieve stable and efficient learning under the uncertainties of maritime steering. We therefore design an algorithm with domain-specific integration and empirical justification. The remainder of this paper is structured as follows: Section 2 discusses relevant work on intelligent collision avoidance for continuous ship steering; Section 3 provides a detailed introduction to the PDPG-based reinforcement learning; Section 4 presents the test results; and Section 5 concludes the paper.

2. Related work

Researchers have made significant progress in addressing challenges in ship steering and collision avoidance. Deraj et al. [14] proposed a deep Q-learning method to solve the path-following problem of ships, using a three-degree-of-freedom dynamic model to describe the ship's motion and training a reinforcement learning agent to interact with the numerical model for waypoint tracking. LADRC has been successfully applied in many control practices, yielding satisfactory results. However, fixed-parameter controllers cannot guarantee optimal control performance for dynamic systems. To enhance the control effect of LADRC through online parameter adjustment, Qin et al. [15] integrated the Deep Deterministic Policy Gradient (DDPG) reinforcement learning algorithm with LADRC for ship heading control, achieving optimal LADRC parameters under varying conditions. Waltz et al. [16] introduced a deep spatiotemporal recurrent neural network architecture for autonomous ship navigation. This architecture can handle an arbitrary number of surrounding target ships while maintaining robustness under partial observability. They also proposed a state-of-the-art collision risk metric to improve the agent's evaluation of different scenarios. Guan et al. [17] addressed the multi-ship collision avoidance problem and the autonomous navigation requirements of Autonomous Surface Ships (ASS). Their proposed multi-ship encounter collision-avoidance decision system combines the Proximal Policy Optimization (PPO) algorithm with the ship domain, accounting for ship manoeuvring characteristics. The reward function for the PPO algorithm ensures that the trained collision-avoidance model complies with COLREGs and prioritizes small-angle steering manoeuvres whenever possible. Sivaraj et al. [18] developed a set of deep reinforcement learning algorithms based on a continuous state-action space for ship path tracking in both calm waters and waves. Their tanker dynamics model accounts for hull, rudder, propulsion, and external wave forces, and the Line of Sight (LOS) forward-looking distance guidance algorithm was employed to calculate tracking and heading errors.

Continuous ship steering requires the vessel's power system to respond swiftly and accurately to steering commands, providing sufficient force and power to execute manoeuvres. This often demands more sophisticated control strategies and adjustment mechanisms. A key consideration in continuous steering is the ship's stability: rapid or frequent changes in direction may lead to issues such as rolling. Ensuring the ship's stability and safety during these manoeuvres requires a comprehensive analysis of its dynamic behaviour [19]. Effective collision avoidance during continuous steering depends on designing an optimal steering strategy—adjusting the ship's angle and speed to navigate different situations safely [20]. This process must account for the ship's power characteristics, manoeuvrability, and environmental influences. In deep reinforcement learning algorithms, agents often overexploit the current best strategy and fail to explore alternative actions, leading to suboptimal collision-avoidance behaviours. Introducing random strategies allows agents to select actions probabilistically rather than deterministically, encouraging exploration of new states and solutions. This probabilistic action selection is critical for addressing continuous control problems. Ship steering control plays a crucial role in ensuring the safety and economic efficiency of navigation. Through simulations of ship steering control and collision avoidance, the ship's motion trajectory can be studied using eligibility traces and random strategies. This approach helps develop optimal collision-avoidance strategies for continuous steering, significantly advancing intelligent collision avoidance in maritime navigation.

Research on intelligent collision avoidance for ship steering can be divided into two main categories: real-world experiments and virtual simulations [21-23]. Modern ships are costly and energy-intensive, making precise collision avoidance experiments in natural waterways prohibitively expensive and inefficient. Additionally, the uncontrollable nature of real-world conditions poses safety risks: even minor navigational errors can lead to serious accidents and significant economic losses. Ships operating in real environments are also subject to complex external factors such as wind and wave interference, resulting in high variability and low repeatability of test scenarios and outcomes. Given these challenges, this study employs virtual simulation software to replicate ship steering and navigation, with a virtual ship agent mimicking the dynamic control system of a real vessel. This approach enables the reconstruction of real-world navigational states and facilitates the training of intelligent collision avoidance algorithms through controlled, repeatable simulation experiments. The primary challenge in applying deep reinforcement learning to intelligent ship collision avoidance lies in abstracting the complexities of the real world [24]. In simulation environments, this challenge is to translate real-world ship steering into a virtual setting that operates independently of natural elements while remaining repeatable and scalable. Specifically, the simulation modelling in this study addresses two key aspects: ship dynamics simulation and waterway environment simulation.

2.1 Ship simulation modeling

This paper focuses on developing intelligent steering control for a model ship to ensure consistency between the agent's actions and the ship's trajectory in both virtual simulations and real-world scenarios. The critical parameters of the model ship are detailed in Table 1.

Table 1 Explanation of table content

Parameters	Values
Length overall of the ship (m)	10
Beam overall of the ship (m)	4
The ship's draught (m)	0.7
Block coefficient	0.56
Rudder angle range (°)	[-35, 35]
The ship's still water speed (m/s)	7.5

The ship's sailing speed is affected by ocean conditions, and this paper accounts for the influence of wind and waves while simplifying these conditions. The impact of irregular wind and waves, denoted as V_a, is estimated using Kwon's approximate method, as shown in Equation (1) [25].

$$\begin{aligned} F_{rou} &= \frac{V_{st}}{\sqrt{g \times L_{pb}}} \\ \frac{\Delta V}{V_{st}} &= \frac{C_\beta C_\mu C_F}{100} \\ \Delta V &= V_{st} - V_a \\ g &= 9.8 \text{ m/s}^2 \end{aligned} \tag{1}$$

where V_{st} represents the ship's still water speed, and L_{bp} is a constant for the same type of ship. ΔV denotes the speed loss caused by ocean weather, while V_a is the ship's sailing speed. The calculation of C_β varies depending on wind angles, as shown in Table 2. The speed loss equation only accounts for involuntary speed reduction, excluding voluntary reductions during critical situations ($B_N > 7$) [26]. The calculation conditions for C_μ and C_F are listed in Tables 3 and 4, respectively.

Table 2 Wind direction attenuation coefficient C_β

Weather direction	Angle	C_β
Head sea and headwinds	0°-30°	$2C_\beta = 2$
Bow sea and bow wind	30°-60°	$2C_\beta = 1.7 - 0.03(B_N - 4)^2$
Beam sea and beam wind	60°-150°	$2C_\beta = 0.9 - 0.06(B_N - 6)^2$
Following sea and following wind	150°-180°	$2C_\beta = 0.4 - 0.03(B_N - 8)^2$

Intelligent collision avoidance in ship steering control leverages artificial intelligence algorithms to enable ships to autonomously detect, assess, and avoid obstacles, ensuring safe navigation. This is primarily achieved through deep reinforcement learning, which allows the ship to perceive its environment, analyze steering movements, make autonomous decisions, and implement control strategies to avoid collisions with other vessels, buoys, or obstacles.

Table 3 Speed attenuation coefficient C_μ

C_b	Ship loading conditions	C_μ
0.55	Normal	$1.7-1.4F_{rou}-7.4(F_{rou})^2$
0.60	Normal	$2.2-2.5F_{rou}-9.5(F_{rou})^2$
0.65	Normal	$2.6-3.7F_{rou}-11.6(F_{rou})^2$
0.70	Normal	$3.1-5.3F_{rou}-12.4(F_{rou})^2$
0.75	Full load or normal	$2.4-10.6F_{rou}-9.5(F_{rou})^2$
0.80	Full load or normal	$2.6-13.1F_{rou}-15.1(F_{rou})^2$
0.85	Full load or normal	$3.1-18.7F_{rou}+28.0(F_{rou})^2$
0.75	Ballast	$2.6-16.3F_{rou}-21.6(F_{rou})^2$
0.80	Ballast	$3.0-16.3F_{rou}-21.6(F_{rou})^2$
0.85	Ballast	$3.4-20.9F_{rou}+31.8(F_{rou})^2$

Table 4 Ship type coefficient C_F

Ship type and loading conditions	C_F
All ships in full load loading condition	$0.5B_N+\dfrac{B_N^{6.5}}{2.7\Delta^2/3}$
All ships in ballast loading condition	$0.7B_N+\dfrac{B_N^{6.5}}{2.7\Delta^2/3}$
ship in normal loading conditions	$0.5B_N+\dfrac{B_N^{6.5}}{2.2\Delta^2/3}$

This approach enhances both the autonomy and safety of ships by considering factors such as ship dynamics, steering angles, environmental conditions, and vessel-specific characteristics.

Ship motion is a complex process, and accurately modeling it is challenging. Simple models cannot fully capture a ship's motion characteristics. This paper focuses on sway, surge, and yaw, and establishes a three-degree-of-freedom MMG model. The hydrodynamic forces and torques acting on the hull, propeller, and rudder will be detailed later. These forces are best described in the ship's fixed coordinate system, as shown in Figure 1. In the figure, G represents the ship's center of gravity, with X and Y denoting the force components along their respective axes, and N the moment around the vertical axis through G. μ and v are the component velocities along the x-axis and y-axis, respectively. ψ represents the heading angle, β the drift angle, δ the rudder angle, and r the rotational velocity. The kinematic equation of the generalized coordinates is given in Equation (2).

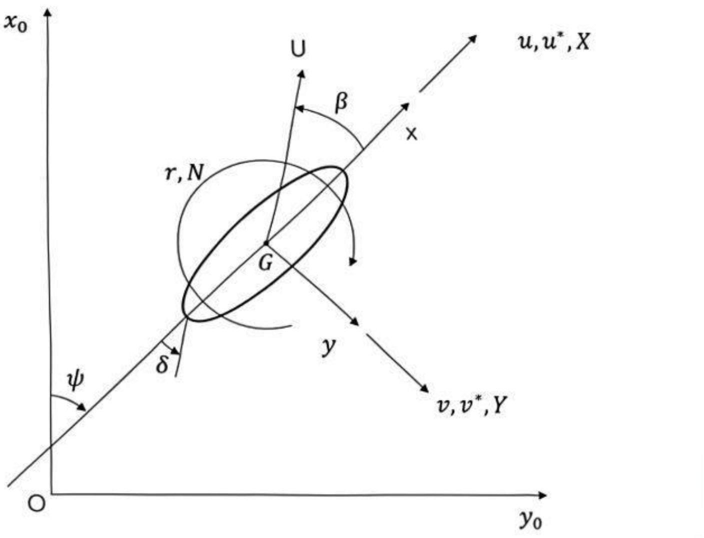


Fig. 1 Ship's steering motion coordinate system

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ r \end{pmatrix} \tag{2}$$

$$\beta = \arctan\left(\frac{v}{u}\right) \tag{3}$$

where the ship's position and heading can be determined from μ, v, and r. The three-degree-of-freedom MMG model is employed to solve these variables, as shown in Equation 4.

$$\begin{cases} (m+m_x)\dot{u} - (m+m_y)ur = X_H + X_P + X_R \\ (m+m_y)\dot{v} + (m+m_x)vr = Y_H + Y_P + Y_R \\ (I_{zz} + J_{zz})\dot{r} = N_H + N_P + N_R \end{cases} \tag{4}$$

where m, m_x, and m_y represent the mass of the ship and its mass components along the x-axis and y-axis. I_{zz} and J_{zz} denote the ship's moment of inertia and additional moment of inertia. $\dot{u}$, $\dot{v}$, and $\dot{r}$ are the accelerations along the x-axis, y-axis, and the yaw acceleration, respectively. The subscripts H, P, and R refer to the hull, propeller, and rudder, respectively, as seen in Equations 5 to 9. Additionally, $X(u)$ is the ship's resistance, and parameters such as X_{vv}, Y_r, and Y_{vv} represent the hydrodynamic parameters.

$$\begin{cases} X_H = X(u) + X_{vv}v^2 + X_{vr}vr + X_{rr}r^2 + X_{vvvv}v^4 \\ Y_H = Y_v v + Y_r r + Y_{vv}vv + Y_{rr}rr + Y_{vvr}v^2 r + Y_{vrr}vr^2 \\ N_H = N_v v + N_r r + N_{vv}vv + N_{rr}rr + N_{vvr}v^2 r + N_{vrr}vr^2 \end{cases} \tag{5}$$

$$\begin{cases} X_P = (1 - t_p)\rho n^2 D_P^4 K_T(J_P) \\ Y_P = 0 \\ N_P = 0 \end{cases} \tag{6}$$

$$K_T(J_P) = a_0 + a_1 J_P + a_2 J_P^2 \tag{7}$$

$$J_P = \frac{u(1 - w_P)}{nD_P} \tag{8}$$

where t_p is a constant, ρ is the density of seawater, and n is the rotation speed of the propeller. J_P is a function related to ω_P in the maneuvering motion. For the convenience of calculation, ω_P is set to a constant. a_0, a_1, and a_2 have correlation coefficients of 0.53, -0.47, and -0.04, respectively.

$$\begin{cases} X_R = (1 - t_R)F_N\sin\delta, \\ Y_R = (1 + a_H)F_N\cos\delta, \\ N_R = (x_R + a_H x_H)F_N\cos\delta, \end{cases} \tag{9}$$

where, F_N and δ are the rudder force and rudder angle, respectively, t_R is the drag reduction of the windward rudder, and a_H is the influence coefficient of steering on the lateral force of the hull. x_R and x_H are the longitudinal coordinates of the steering rudder and the distance to the center of the lateral force of the hull, respectively. This paper is based on relevant literature [27], and a model ship is modeled based on a particular model ship. The main parameters are shown in Table 5.

Table 5 Hydrodynamic parameters and derivatives used in the maneuvering simulations

Parameters	Values	Parameters	Values
m_x	0.050	Y_{uv}	0.593
m_y	1.034	Y_{rr}	0.342
I_{zz}	1.820	Y_{uvr}	−0.483
J_{zz}	0.063	Y_{urr}	0.834
X_{vv}	−0.055	N_v	0.111
X_{vr}	−0.018	N_r	−0.047
X_{rr}	−0.012	N_{vv}	−0.053
X_{vivvv}	−0.042	N_{rr}	0.0214
Y_v	0.225	N_{vur}	−0.617
t_p	0.220	N_{vrr}	0.051

2.2 Ship steering control collision avoidance simulation environment

This paper focuses on ship navigation within curved channels and establishes a ship steering control and collision avoidance environment tailored to the unique characteristics of these channels. To address more complex scenarios, collision avoidance environments with varying bending angles are developed. The navigation diagram of the simulation environment is shown in Figure 2. Specifically, the channel implementation is based on previous research [28]. In Figure 2, the yellow figure represents the ship agent (controlled ship), the blue figure represents other ships (target ships), and the space between the two white solid lines indicates the ship's navigable area. The depth of the blue background reflects varying marine weather conditions, with meteorological details illustrated in Figure 3. The horizontal and vertical coordinates display longitude and latitude, representing the meteorological conditions at different locations. Darker colors correspond to higher meteorological values, which affect ship navigation across various times and regions. In this simulation, the ship agent must learn two key control tasks: navigating through the complex channel and avoiding collisions with other vessels. The coordinate system, set at 600x600 pixels, can be converted to geographic coordinates using the Frenet framework [29].

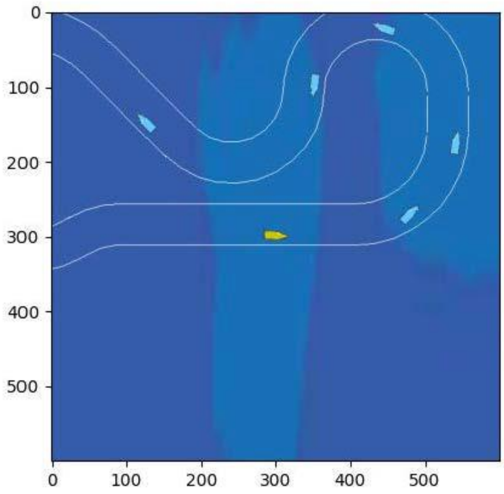


Fig. 2 Schematic diagram of the simulation environment

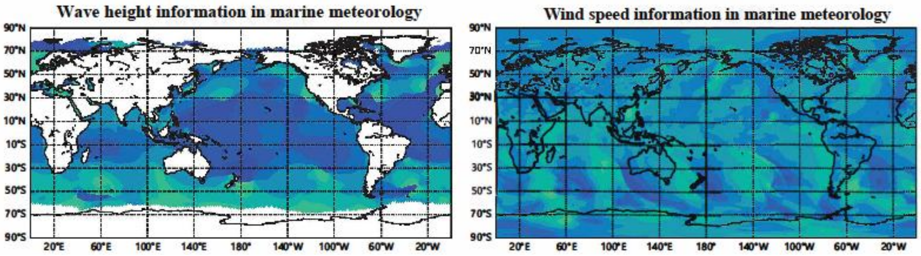

Wave height information in marine meteorology
Wind speed information in marine meteorology
90°N
70°N
50°N
30°N
10°N
10°S
30°S
50°S
70°S
90°S
20°E
60°E
100°E
140°E
180°
140°W
100°W
60°W
20°W

Fig. 3 The marine meteorological information

2.3 Calculate the risk of ship collision

This paper employs the fuzzy mathematics method [30] to evaluate the collision risk associated with ship steering. The risk index between two vessels during steering is determined by their relative motion parameters, including the distance to the closest point of approach (DCPA), the time to the closest point of approach (TCPA), the relative distance between the ships (Dis), the relative position (α_{rel}), the relative speed (v_{rel}), and the direction of the relative speed (ψ_{rel}). The calculations for DCPA and TCPA are provided in Equation 10.

$$\begin{cases} DCPA = Dis \cdot \sin(\psi_{rel} - \alpha_{rel} - \pi), \\ TCPA = Dis \cdot \cos(\psi_{rel} - \alpha_{rel} - \pi)/v_{rel} \end{cases} \tag{10}$$

$$Dis = \sqrt{(x_{tar} - x_{own})^2 + (y_{tar} - y_{own})^2} \tag{11}$$

where (x_{tar}, y_{tar}) and (x_{own}, y_{own}) represent the positions of the target ship and the own ship, respectively, during steering. Fuzzy logic effectively captures knowledge and experience with imprecise boundaries, making it a suitable method for assessing collision risks in ship steering. To simplify calculations, this paper focuses on the distance to the closest point of approach (DCPA) and the time to the closest point of approach (TCPA) as the primary factors in evaluating collision risk. The collision risk index (CRI) is then calculated as follows:

$$col_{CRI} = p \cdot col_{DCPA} + (1 - p) \cdot col_{TCPA} \tag{12}$$

$$col_{DCPA} = \begin{cases} 0 \\ \frac{1}{2} - \frac{1}{2}\sin\left[\frac{\pi}{dis_2 - dis_1}\left(|DCPA| - \frac{dis_1 + dis_2}{2}\right)\right] \\ 1 \end{cases} \tag{13}$$

where $dis_2 < |DCPA|$ indicates that $col_{DCPA} = 0$, while $|DCPA| \leq dis_1$ signifies that $col_{DCPA} = 1$. For other scenarios, col_{DCPA} is expressed in Equation (13). The values of dis_1 and dis_2 are calculated as shown in Equations (14) and (15).

$$dis_1 = \begin{cases} 1.1 - \dfrac{0.2 \times \psi_{rel}}{\pi} & 0^\circ \leq \psi_{rel} < 112.5^\circ \\ 1.0 - \dfrac{0.4 \times \psi_{rel}}{\pi} & 112.5^\circ \leq \psi_{rel} < 180^\circ \\ 1.0 - \dfrac{0.4 \times (2\pi - \psi_{rel})}{\pi} & 180^\circ \leq \psi_{rel} < 247.5^\circ \\ 1.1 - \dfrac{0.2 \times (2\pi - \psi_{rel})}{\pi} & 247.5^\circ \leq \psi_{rel} \leq 360^\circ \end{cases} \tag{14}$$

$$dis_2 = K \times dis_1 \tag{15}$$

where d_1 represents the minimum distance between the two ships, while d_2 denotes the safe encounter range. The coefficient K is set to *1.8*. The calculation of r_{TCPA} is detailed in Equation (16).

$$col_{TCPA} = \begin{cases} 0, & time_2 < |TCPA| \\ \left(\dfrac{time_2 - |TCPA|}{time_2 - time_1}\right)^2, & time_1 < |TCPA| \leq time_2 \\ 1, & 0 \leq |TCPA| \leq time_1 \end{cases} \tag{16}$$

$$time_1 = \begin{cases} \dfrac{\sqrt{(Dis_1^2 - DCPA^2)}}{DCPA \leq D_1} & DCPA \leq D_1 \\ \dfrac{Dis_1 - DCPA}{v_{rel}} & DCPA > D_1 \end{cases} \tag{17}$$

$$time_2 = \begin{cases} \dfrac{\sqrt{(Dis_2^2 - DCPA^2)}}{Dis_2 - DCPA} & DCPA \leq Dis_2 \\ \dfrac{v_{rel}}{v_{rel}} & DCPA > Dis_2 \end{cases} \tag{18}$$

where $time_1 = 3$ min indicates the time at which the two ships collide, while $time_2 = 10$ min marks the time when the target ship is observed. Dis_1 and Dis_2 represent the ship's closest operating distance and safety distance, respectively.

2.4 Regulations for Preventing Collisions at sea

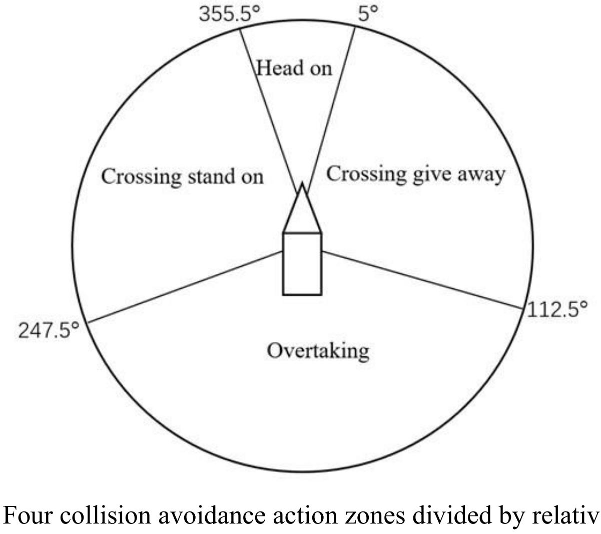


Fig. 4 Four collision avoidance action zones divided by relative position

COLREGs classify ship encounters based on their relative positions and outline the rules that each vessel must follow to avoid collisions. From a first-person perspective, the reference vessel used to determine navigation direction is called the own ship (OS). At the same time, the dynamic obstacle is the target ship (TS), which poses the most significant hazard. When the own ship encounters the target ship, it must either maintain its current course or execute collision avoidance maneuvers in accordance with COLREGs. Before addressing the intelligent collision-avoidance decision problem for dynamic obstacles using deep reinforcement learning, it is essential to consider these maritime collision-avoidance rules. COLREGs are mandatory regulations established by the International Maritime Organization to enhance ship safety and minimize collisions, outlining various encounter scenarios and their corresponding avoidance directions. Therefore, intelligent collision avoidance for dynamic obstacles must adhere to COLREG rules to ensure safe maritime navigation [31]. According to COLREGs, the relative positions of ship encounters are categorized into four obstacle avoidance strategy areas, as illustrated in Figure 4.

2.5 Ship steering control strategy

During navigation, ships must adjust their course according to the defined route, which often includes passing through curved channels. In these areas, ships must execute turns or change course to avoid collisions with other vessels or obstacles, ensuring a safe, smooth passage. Various factors influence turning strategies in such channels, including the ship's characteristics, environmental conditions, and weather, all of which require different control methods for optimal maneuverability. The traditional turning strategy relies on adjusting both the rudder angle and engine thrust. Generally, a larger rudder angle produces a smaller turning radius and reduces turning time. While this method is intuitive and straightforward, it is heavily influenced by the ship's characteristics—such as size and handling—as well as external environmental factors, including wind, waves, and currents. These influences can significantly impact turning accuracy, often rendering the traditional approach less reliable in complex or adverse conditions.

The model predictive control (MPC) method advances traditional approaches by combining predictive strategies with control techniques. By continuously analyzing the ship's current state and motion dynamics, MPC develops optimal control strategies to ensure stable heading during turns. Although this method requires significant computational power and relies on complex algorithms, it provides precise and efficient control over the ship's navigation, even in challenging conditions. The artificial intelligence control method leverages machine learning and deep reinforcement learning to process extensive datasets, enabling the vessel to adjust its control parameters automatically. This approach allows for high-precision, adaptive turning control as the system learns and refines its performance over time. It is particularly well-suited to dynamic environments and uncertain scenarios, offering real-time navigation capabilities. The AI-driven method is especially advantageous in complex situations where traditional and predictive methods may fall short due to environmental uncertainties or rapidly changing conditions.

3. Intelligent collision avoidance method for continuous steering of ships based on improved PDPG

Most existing DRL ship-collision studies are demonstrated in comparatively more straightforward navigation settings and often do not account for the changing continuous action space caused by variations in ship speed and steering. Our work explicitly targets continuous steering in curved channels, motivated by real curved waterways and their constrained maneuvering space and dynamic conditions. Unlike methods that discretize maneuvers or limit control to heading-only, we model the control decision as a continuous action vector that includes both acceleration and rudder angle, enabling smoother, more realistic ship steering behavior. Beyond applying a standard DRL algorithm, we propose an improved PDPG-based framework that (i) optimizes the policy network structure to strengthen representation capacity and (ii) introduces an adaptive exploration rate plus dynamic balancing of random strategies to handle better exploration–exploitation in continuous action spaces, reducing premature convergence and improving robustness for continuous steering. We emphasize that the policy is learned in an environment grounded in the MMG maneuvering model, and the framework evaluates steering performance via an explicit evaluation mechanism, rather than relying on hand-designed rule-only steering logic. The main differences between the improved PDPG method used in this paper and the existing baseline method are shown in Table 6.

Table 6 Differences between the improved PDPG algorithm and the baseline algorithm

Method	Policy type	Action type	Exploration	Replay
Standard actor–critic	Off-policy	Deterministic	Depends on implementation	Optional
DDPG	Off-policy	Deterministic	Action noise	Uniform
TD3	Off-policy	Deterministic	Action noise + policy smoothing	Uniform
PPO	On-policy	Stochastic	Inherent stochastic policy	Typically
Improved PDPG	Off-policy	Actor-critic	Diagonal Gaussian+adaptive exploration	Prioritized replay

Path planning provides a reference route centerline. Steering control is the continuous control variable applied to the ship. Collision avoidance is the decision objective integrated into the RL policy. At each step, the agent selects continuous steering commands that simultaneously maintain progress on the planned channel and avoid collisions. Reinforcement learning enables an agent to optimize its behavior through continuous interaction with the environment [32, 33]. The overall framework diagram is shown in Figure 5. Upon executing an action in a specific state, the agent receives a scalar reward that reflects the effectiveness of the action. The primary objective of the agent is to discover a policy that maximizes cumulative rewards, which are estimated using the value function. This value function allows the agent to leverage its past experiences to inform and guide future decision-making.

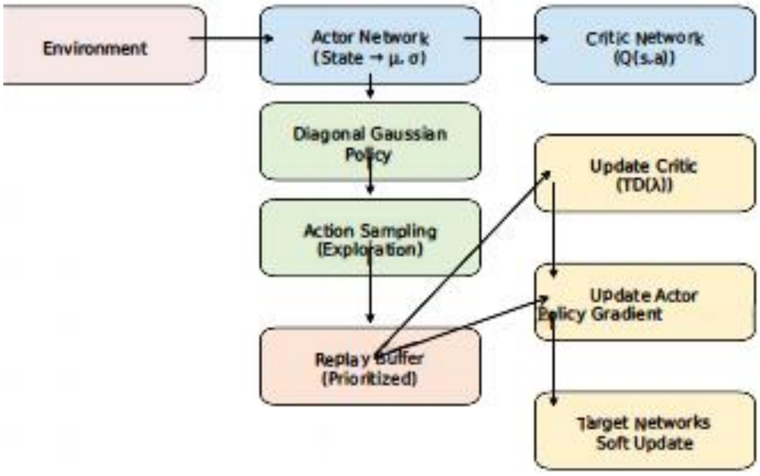

Environment
Actor Network (State → μ, σ)
Critic Network (Q(s,a))
Diagonal Gaussian Policy
Action Sampling (Exploration)
Replay Buffer (Prioritized)
Update Critic (TD(λ))
Update Actor Policy Gradient
Target Networks Soft Update

Fig. 5 The framework of improved PDPG algorithm

With the continued advancement of reinforcement learning, deep reinforcement learning (DRL) algorithms that incorporate deep learning techniques have gained prominence. One notable example is the Pathwise Derivative Policy Gradient (PDPG) algorithm, a specialized actor-critic method designed to tackle the complexities of reinforcement learning in continuous action spaces. Reinforcement learning approaches are typically classified into three categories [34]: actor-only methods, critic-only methods, and actor-critic methods.

Actor-only methods generally use a parameterized policy to generate continuous actions. However, these methods, which often rely on policy gradient techniques, tend to suffer from high variance in their gradient estimates. On the other hand, critic-only methods utilize temporal difference learning, leading to lower variance in the estimation of expected returns [35].

A simple way to derive a policy in critic-only methods is by using a greedy approach, where the value function selects actions that maximize the expected return. This involves applying a greedy policy at each encountered state to determine the action that yields the highest return. However, this process can be computationally demanding, particularly in continuous action spaces. As a result, critic-only methods often resort to discretizing the action space, turning the optimization problem into an enumeration task. This limitation reduces their effectiveness in handling continuous action problems and may impede the discovery of optimal actions.

Actor-critic methods combine the advantages of both actor-only and critic-only approaches. The parameterized actor allows the agent to calculate continuous actions directly, bypassing the need for value function optimization. Meanwhile, the critic provides an estimate of expected returns, enabling the agent to update its policy using gradients with reduced variance, which speeds up the learning process. This combination typically leads to better convergence compared to critic-only methods.

3.1 The PDPG algorithm

The PDPG algorithm is a specialized form of the actor-critic algorithm. The value function is parameterized by a vector $\theta \in \mathbb{R}^q$ and represented as either $V_\theta(x)$ or $Q_\theta(x,a)$. In the case of a linear parameterization, the basis function is denoted by ϕ, as described in Equation (19).

$$V_\theta(x) = \theta^\top \phi(x) \; or \; Q_\theta(x,a) = \theta^\top \phi(x,a) \tag{19}$$

where the random policy π is parameterized by $\vartheta \in \mathbb{R}^q$ and is expressed as $\pi_\vartheta(x,\mu)$. When the policy is represented by $\pi_\vartheta(x)$, it corresponds to a deterministic policy, meaning that it no longer represents a probability density function but rather a direct mapping from states to actions, where $\mu = \pi_\vartheta(x)$.

The actor-critic algorithm seeks to determine the optimal policy for a given, fixed Markov Decision Process (MDP). A key requirement for this is that the critic must accurately evaluate the current policy. In essence, the critic's objective is to approximate the solution to the Bellman equation for that policy. The difference between the left-hand and right-hand sides of the Bellman equation, whether in a discounted or average reward setting, is referred to as the Temporal Difference (TD) error. This TD error is used to update the critic. The critic's function approximates the policy using transition samples (x_k, a_k, r_{k+1}, x_{k+1}), and the TD error estimate is defined in Equation (20).

$$\delta_k = r_{k+1} + \gamma V_{\theta_k}(x_{k+1}) - V_{\theta_k}(x_k) \tag{20}$$

The standard way to update the critic is to update the TD error using gradient descent, as shown in the Equation (21).

$$\theta_{k+1} = \theta_k + \alpha_{c,k}\delta_k \nabla_\theta V_{\theta_k}(x_k) \tag{21}$$

where $\alpha_{c,k}$ is the critic's learning rate. The linear parameterized function approximation formula Equation (19) can be simplified as shown in Equation (22).

$$\theta_{k+1} = \theta_k + a_{c,k}\delta_k(x_k) \tag{22}$$

This Temporal Difference (TD) method is also known as $\mathrm{TD}(0)$ learning but does not incorporate eligibility traces [36]. In this paper, we extend the TD method to include eligibility traces, resulting in the $TD(\lambda)$ method. When Equation (20) is used to update the critic, the agent receives a single-step reward; however, this reward often reflects the cumulative outcome of multiple steps. Eligibility traces offer a more effective way to assign credit to states or state-action pairs that have been visited over several steps. The eligibility trace vector, representing all q features at time k, is denoted as $z_k \in \mathbb{R}^q$, and its update equation is provided in Equation (23).

$$z_k = \lambda\gamma z_{k-1} + \nabla_\theta V_{\theta_k}(x_k) \tag{23}$$

The probability of selecting action a in state s is determined based on agent-environment interactions, after which the cumulative reward is calculated following the execution of action a until the end of the episode. The policy parameter θ is then updated according to Equation (24).

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}\left(\sum_{t'=t}^{T_n}\gamma^{t'-t}\, r_{t'}^n - b\right)\nabla \mathrm{log} p_\theta(a_t^n \mid s_t^n) \tag{24}$$

where $\sum_{t',-t}^{I_n} \gamma^{t'-t}\, r_{t'}^{n} - b$ represents the accumulated reward from time t to T_n, where γ is the discount factor. G represents the cumulative reward. Due to the stochastic nature of the interaction process, G can be unstable. To ensure stability during training, the expected value of G is used in place of the sampled value. The expected value of G is given in Equation (25).

$$\mathbb{E}[G_t^n] = Q_{\pi_\theta}(s_t^n, a_t^n) \tag{25}$$

If $\sum_{t'=t}^{T_n} \gamma^{i'-t}\, r_{t'}^{n} - b$ is represented by $\mathbb{E}[G_t^n]$, then the actor and critic are integrated. The value function of the actor-critic algorithm is provided in Equation (26).

$$\begin{aligned} J(\pi) &= E\left\{\sum_{k=0}^{\infty} \gamma^k\, r_{k+1} \mid d_0, \pi\right\} \\ &= \int_S d_r^\pi\,(s) \int_A \pi\,(s, \alpha) \int_S f\,(s, a, s')\rho(s, a, s')ds'ds \end{aligned} \tag{26}$$

where s represents the state, a denotes the action, $d_r^\pi(s)$ represents the state distribution under policy π, and ρ represents the probability density function.

In intelligent collision avoidance decision-making for ship steering control, evaluating the merits of individual steering actions alone is insufficient. It is equally important for stakeholders to identify actions that optimize overall steering control. The basic actor-critic algorithm enables the actor network to generate action values, providing insights into the relative advantages and disadvantages of each action. However, it does not specify how to execute these actions to achieve optimal outcomes.

According to related research [37], to identify actions that maximize the action value, an actor is designed with the state as its input and the expected action as its output. This action is then passed to the critic network, which seeks to maximize the objective function $Q_\pi(s, a)$, as defined in Equation (27).

$$\pi'(s) = \underset{a}{\operatorname{argmax}} Q_\pi(s, a) \tag{27}$$

In the training scenario, the critic is implemented as a neural network (Q-network) that takes the ship's current state and the chosen action as inputs to produce a corresponding value. As the actor is updated iteratively, it uses the ship's state as input to generate a steering action, which is then fed into the critic, with the aim of achieving the highest expected output. During parameter updates, the critic's parameters remain fixed, while only the actor's parameters are adjusted. The goal of applying the gradient ascent method is to maximize the critic's output.

In continuous steering collision avoidance, constant or poorly tuned exploration can lead to premature convergence to local optima. Our improved PDPG introduces an adaptive exploration-rate schedule and a random strategy mechanism to dynamically balance exploration and exploitation, explicitly motivated by the need for continuous steering and the avoidance of local optima. We further employ a diagonal Gaussian policy in a continuous action space to avoid over-exploiting the current best strategy and improve robustness during learning. Ship collision-avoidance training in simulation generates many trajectories; off-policy methods can reuse them efficiently. Our improved PDPG uses prioritized experience replay to focus learning on informative transitions, improving sample efficiency and convergence quality. Nonlinearity, time-varying parameters, and disturbances/uncertainty explicitly characterize the ship steering control problem. We position improved PDPG as a practical learning-based controller to address unknown disturbances, ship dynamics, and nonlinear control in the steering model. The definitions of state, action, and reward are as follows: the state represents the ship agent's observations within the continuous steering collision avoidance simulation environment, serving as the foundation for intelligent collision avoidance in steering control. In this study, the state includes the ship's position, speed, and heading information. The state space is defined as:

$$State = (S_{own}, S_{tar}) \tag{28}$$

$$S_{own} = (ex_{own}, x_{own}, y_{own}, v_{x_{own}}, v_{y_{own}}, \sin\delta_{own}) \tag{29}$$

$$S_{tar} = (ex_{tar}, \Delta x_{to}, \Delta y_{to}, \Delta v_{xto}, \Delta v_{yto}, \sin\delta_{tar}) \tag{30}$$

where $ex_{own} \in 0{,}1$ and $ex_{tar} \in 0{,}1, -1 < \Delta x_{to} < L$, and $t \le 5$. $\mathrm{S_o}$ denotes the own ship, while $\mathrm{S_t}$ refers to nearby target ships. The variable ex indicates the presence of a ship, where 1 signifies that a ship is present and 0 indicates otherwise. x and y represent the ship's navigational position. $v_{x_{own}}$ and $v_{y_{own}}$ denote the lateral and longitudinal velocity components of the ship, respectively. The variable δ_{own} represents the vessel's heading angle, while Δ denotes a relative value. l is the length of the steering ship, and L represents the maximum observation distance for the own ship. Finally, t represents the number of surrounding ships detected by the own ship, with lateral and longitudinal motions controllable through steering and speed in the motion space.

Action refers to the decisions made by the ship to execute the steering control collision avoidance task. In this study, actions consist of the ship's acceleration α and rudder angle δ. The action space is defined as:

$$Actions = (\alpha, \delta), \alpha \in [-3{,}3]\ \mathrm{m/s^2}, \delta \in (-35°, 35°) \tag{31}$$

The quality of the actions taken by the ship agent is evaluated using rewards, which take into account four aspects:

1. Safety means avoiding collisions with other ships during navigation.
2. Efficiency means keeping the ship's speed within a specific speed range.
3. Comfort means turning the ship's heading angle within a specific range.
4. Rules mean sailing on the center line of the ship's channel.

Our evaluation of rewards primarily includes collision reward R_{col}, efficiency reward R_{eff}, comfort reward R_{com}, and rule-compliance reward R_{rul} [38]. The rewards are defined as:

$$Reward = R_{col} + R_{eff} + R_{com} + R_{rul} \tag{32}$$

$$R_{col} = k_1 \tag{33}$$

$$R_{eff} = M \cdot \frac{V - V_{\min}}{V_{\max} - V_{\min}} \tag{34}$$

$$R_{com} = k_2 \tag{35}$$

$$R_{rul} = k_3 \tag{36}$$

where V, $V_{\min}$, and $V_{\max}$ represent the navigation speed, the lower limit, and the upper limit of the speed, respectively. k_1, k_2, and k_3 represent constant values in the calculation formula, and m represents the correlation coefficient. In this paper, $k_1 = -200$, $k_2 = -5$ and $k_3 = 10$.

3.2 PDPG steering collision avoidance algorithm based on adaptive exploration rate and random strategy.

The adaptive exploration rate dynamically adjusts ω over the training process to maintain an effective balance between exploration (early stage) and exploitation (later stage). The ω can be calculated using Equation (37).

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min})e^{-\lambda t} \tag{37}$$

where ω is the exploration variance in iteration t, $\omega_{\max}$ and $-\omega_{\min}$ are the upper and lower bounds for exploration variance, λ is the decay coefficient controlling the speed of convergence.

In the study of intelligent collision avoidance methods for continuous ship steering, the choice of constant steering actions significantly influences the outcomes. The ship's intelligent agent must balance exploration—testing new, unfamiliar actions to discover improved strategies—and exploitation—selecting the best-known actions based on past experience. Incorporating a random strategy enhances exploration, increasing the likelihood of discovering the optimal strategy while preventing premature convergence to local optima. Additionally, this random strategy improves the algorithm's stability, leading to a smoother gradient descent process and enhancing both convergence and robustness.

In deep reinforcement learning, stochastic policies include categorical and diagonal Gaussian policies. Categorical policies are typically used in discrete action spaces, while diagonal Gaussian policies are employed in continuous action spaces. A diagonal Gaussian policy is a specific multivariate Gaussian distribution where the covariance matrix is diagonal. It is characterized by a mean vector (μ) and a covariance matrix (Σ). Diagonal Gaussian policies use a neural network to map observations to the mean action ($\mu_\theta(s)$). The covariance matrix is usually implemented in two ways: one uses a log-standard deviation vector ($\log\sigma$), which is an independent parameter not dependent on the state, while the other uses a neural network that maps the state to the log-standard deviation ($\log\sigma_\theta(s)$), potentially sharing parameters with other networks.

During the sampling process, for a given mean action ($\mu_\theta(s)$) and standard deviation ($\sigma_\theta(s)$), a distribution z, which follows a spherical Gaussian ($z \sim N(0, I)$), is introduced. The action sample can then be calculated using Equation (38).

$$a = \mu_\theta(s) + \sigma_\theta(s) \odot z \tag{38}$$

where $\odot$ represents the element-wise product of two vectors.

For a diagonal Gaussian distribution with mean $\mu = \mu_\theta(s)$ and standard deviation $\sigma = \sigma_\theta(s)$, the log-likelihood of a k-dimensional action a is given in Equation (39).

$$\log\pi_\theta(a \mid s) = -\frac{1}{2}\left(\sum_{i=1}^{k}\left(\frac{(a_i - \mu_i)^2}{\sigma_i^2} + 2\log\sigma_i\right) + \log 2\pi\right) \tag{39}$$

Based on the analysis, the improved PDPG algorithm is utilized to tackle the problem of intelligent collision avoidance in continuous ship steering. The detailed process for applying the improved PDPG to this problem is presented in Algorithm 1.

In the intelligent collision avoidance algorithm for continuous ship steering based on the improved PDPG, the ship's agent obtains training samples (s_t, a_t, r_t, s_{t+1}) through interactions with the simulation environment. The agent is trained using prioritized experience sampling. In line 9 of the algorithm, a diagonal Gaussian policy is adopted to prevent the ship's agent from over-exploiting the current best strategy, which could cause the continuous steering collision avoidance action to converge to a local optimum. The algorithm runs for D steps to update the parameters of both the critic and actor networks, ensuring the algorithm can learn the optimal collision avoidance actions. Lines 7-12 of Algorithm 1 represent the core calculation process of the improved algorithm, guiding the ship's agent to learn various continuous steering collision avoidance actions necessary for successful collision avoidance.

Algorithm 1 Application of improved PDPG in ship continuous steering intelligent collision avoidance

```
Input: Ship navigation situation
Output: $s_t$, $a_t$, $r_t$, $s_{t+1}$
(1) $Shipnavigationsituation, InitializecriticC, targetcriticC, actor\pi,$
    $AdaptiveExplorationRatePadandtargetactor\pi = \pi, epoch_{n}um;$
(2) Calculate collision risk index based on DCPA and TCPA functions;
(3) Judgment of the encounter between two ships;
(4) for e = 1to epoch$_{num}$;
(5) Repeat;
(6) According to the interaction between the ship agent and the environment, ($s_t$, $a_t$, $r_t$, $s_{t+1}$) is generated
    and stored in the playback buffer pool;
(7) Obtain Get the motion trajectory from the playback buffer pool based on the priority experience sampling
    ($s_i$, $a_i$, $r_i$, $s_{i+1}$);
(8) TD($\lambda$) is used to update the parameter C;
(9) Diagonal Gaussian policies select the ship's continuous steering collision avoidance action.;
(10) Update the parameters of actor $\pi$ with the goal of ensuring the maximum $C(s_i, \pi(s_i))$;
(11) Update $\hat{C}$= C after running step D;
(12) Update $\hat{\pi} = \pi$ after running step D;
(13) end for;
```

(14) return (s_t, a_t, r_t, s_{t+1});

4. Experiment and discussion

This section evaluates the effectiveness of the improved PDPG algorithm in solving the intelligent collision avoidance decision-making problem for ship steering control through a simulation experiment focused on collision avoidance.

4.1 Experimental running environment and parameter settings

The basic parameters of the ship steering control collision avoidance simulation environment are presented in Table 7. This section compares the improved PDPG algorithm with other reinforcement learning algorithms, including Deep Deterministic Policy Gradient [39], Proximal Policy Optimization (PPO) [40], and Twin Delayed DDPG (TD3) [41]. The key parameters for these algorithms are outlined in Table 8, with additional settings drawn from relevant literature. The navigation diagram of the ship steering control simulation environment is illustrated in Figure 2. During the simulation, other ships are in motion, necessitating adjustments to the rudder angle and speed to avoid collisions.

Table 7 Experimental environment

Components	Attribute
Operating system	CentOS Linux release 7.6.1810 (Core)
Memory	754G
CPU	Intel(R) Xeon(R) Platinum 8260 CPU
Basic frequency	2.40GHZ
Programming language	Python 3.8.3
Graphics card	NVIDIA Corporation TU102GL (rev a1)

Table 8 The key parameters of the comparative algorithm for the steering control

Algorithm	Policy	Learning-rate	Gamma
the improved PDPG	MLpPolicy	3.5e-4	0.99
DPDG	MLpPolicy	3.5e-4	0.99
PPO	MLpPolicy	2.5e-4	0.99
DDPG	MLpPolicy	3e-4	0.99
TD3	MLpPolicy	2e-4	0.99

One ship agent operates in the simulation environment alongside three other ships. The controlled ship and the other ships are initialized with specific speeds. As described in Section 2, ocean weather conditions influence changes in the ship's speed. The initial speed settings for the ships are provided in Table 9, with the range of speed variation being [0, 15]. The other ships are assumed to move forward along the centerline of the channel in the simulation environment. The parameters for the ship steering dynamics model are set as $\mathrm{n}_3 = 10^{-2}$, $\mathrm{n}_1 = 10^{-3}$, and $\mathrm{k} = 1.0$.

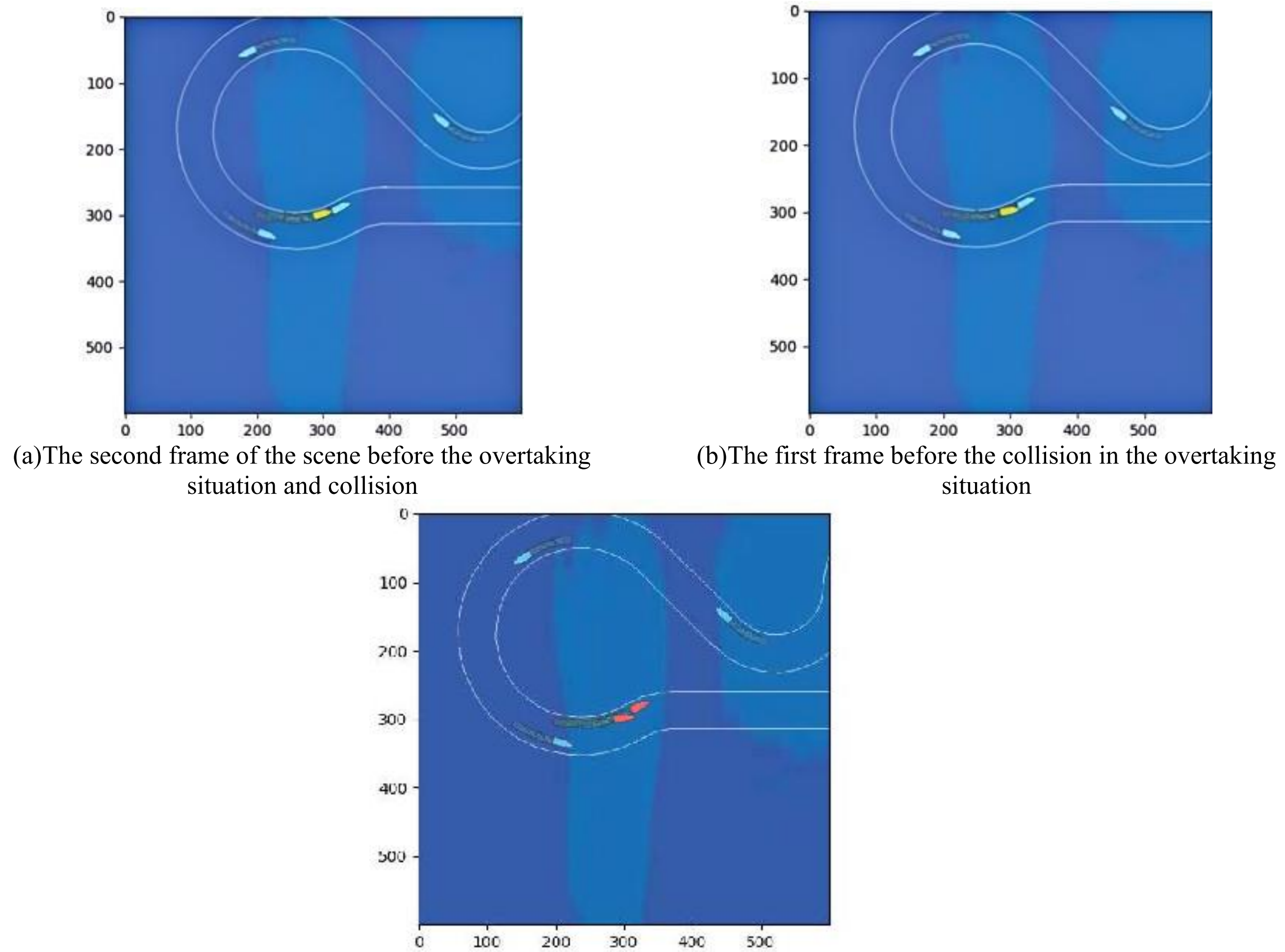
Table 9 Parameters for configuring moving objects in a steering control simulation environment

ship	x(m)	y(m)	v_x(m/s)	v_y(m/s)	δ
ego-ship	13.00	3.00	7.00	0	0
ship 1	12.00	5.00	6.50	0	0
ship 2	15.00	7.00	7.40	0	0
…	…	…	…	…	…
ship n	11.00	6.00	8.7	0	0

4.2 Decision analysis of continuous steering intelligent collision avoidance

Using the improved PDPG algorithm, the ship agent's steering control collision avoidance decision is trained in a continuous steering control collision avoidance simulation. Since the simulation process is both dynamic and constant, to clearly demonstrate the collision avoidance process, the actions during the simulation are sampled and displayed at fixed intervals. The collision avoidance decisions were derived from 30,000 simulation experiments conducted with the ship agent, learning from the simulation samples based on the improved PDPG algorithm. This section presents two or three consecutive frames of the ship agent during the collision avoidance process. In the continuous steering collision avoidance simulation environment, the ship agent is scaled down according to the model ship described in Chapter 3 and operates within the simulation environment.

This section primarily demonstrates the process of training ship agents to avoid collisions in a continuous steering collision avoidance simulation using the improved PDPG algorithm. Figure 6 depicts a scene of a ship collision within the steering control collision avoidance simulation environment. The collision process unfolds as follows: Ocean weather influences cause fluctuations in the ship agent's speed. As the simulation progresses, the ship agent overtakes other ships, as illustrated in Figure 6a. According to the CRI curve in Figure 7, the ship agent must adjust its rudder angle and speed to avoid a collision. Figure 6b shows the ship changing its heading, but due to errors in controlling speed and rudder, the ship agent collides with other vessels ahead, as shown in Figure 6c.



(a)The second frame of the scene before the overtaking situation and collision

(b)The first frame before the collision in the overtaking situation

(c)Overtaking situation collision scene

Fig. 6 Overtaking collision chart in a steering control simulation environment

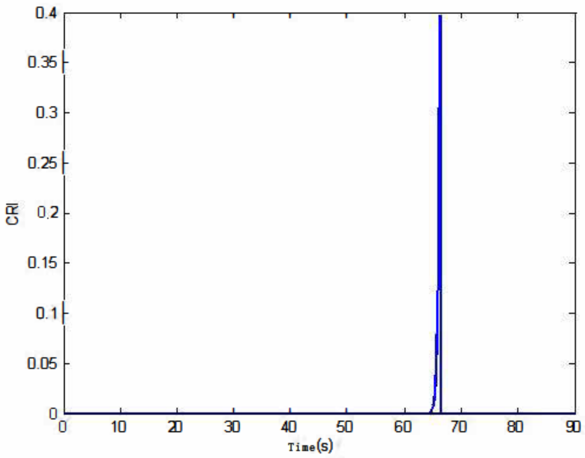
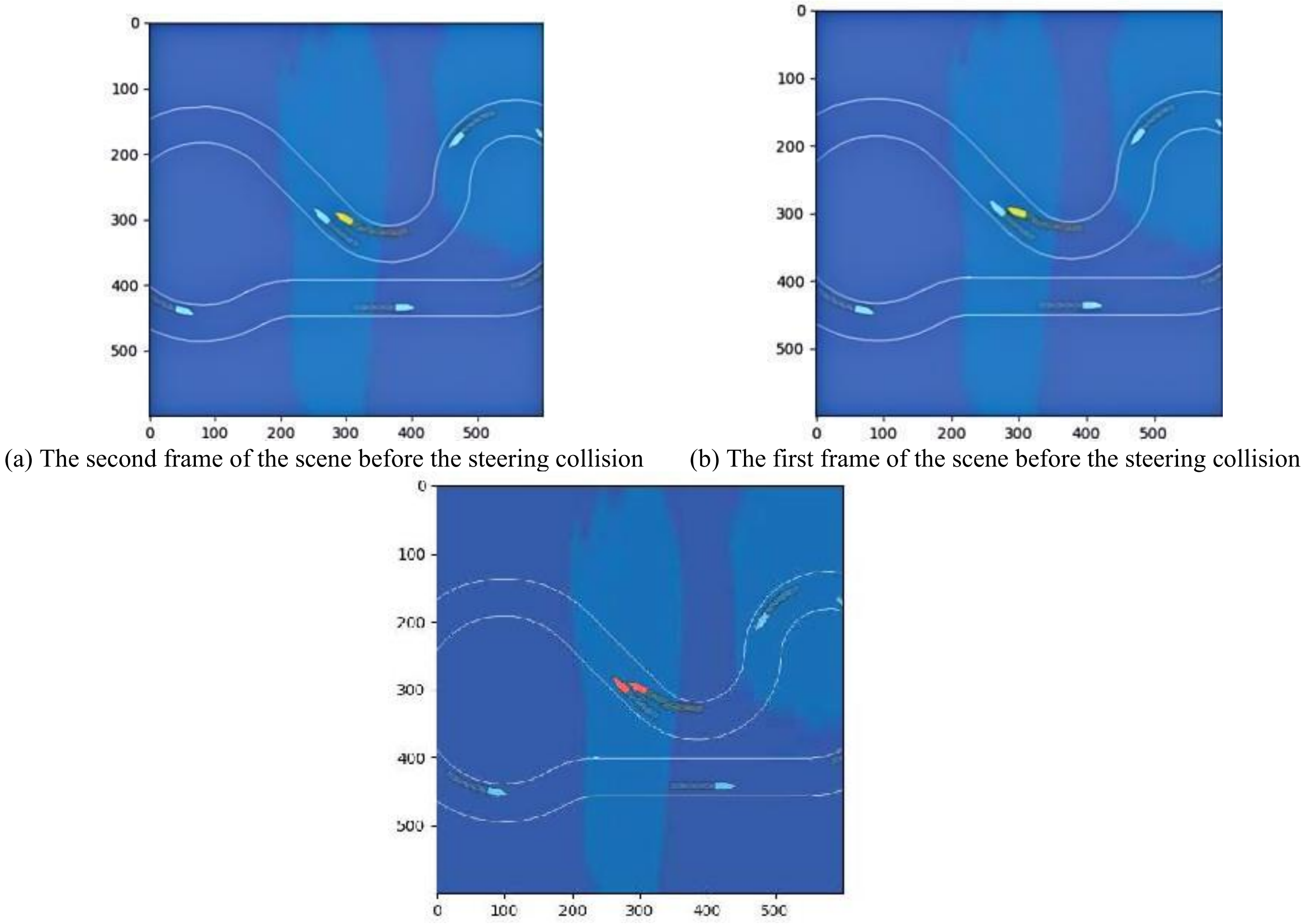


Fig. 7 The CRI curve in a steering control simulation environment

Figure 8 illustrates another scenario of a ship collision in a steering control collision avoidance simulation. Using the improved PDPG algorithm, the ship agent needs to explore new rudder angle controls. In this scenario, the ship agent is sailing on the right side of other ships. Based on the collision risk curve, the agent performs a left turn. However, as the ship's speed changes and it remains on the right side of other ships, the agent moves toward them, as shown in Figures 8a and 8b. Meanwhile, the other ships continue to follow the ship channel without adjusting their rudder angles. Consequently, in the following state, the ship agent collides with the other ships, and both vessels turn red in the simulation environment, as depicted in Figure 8c.



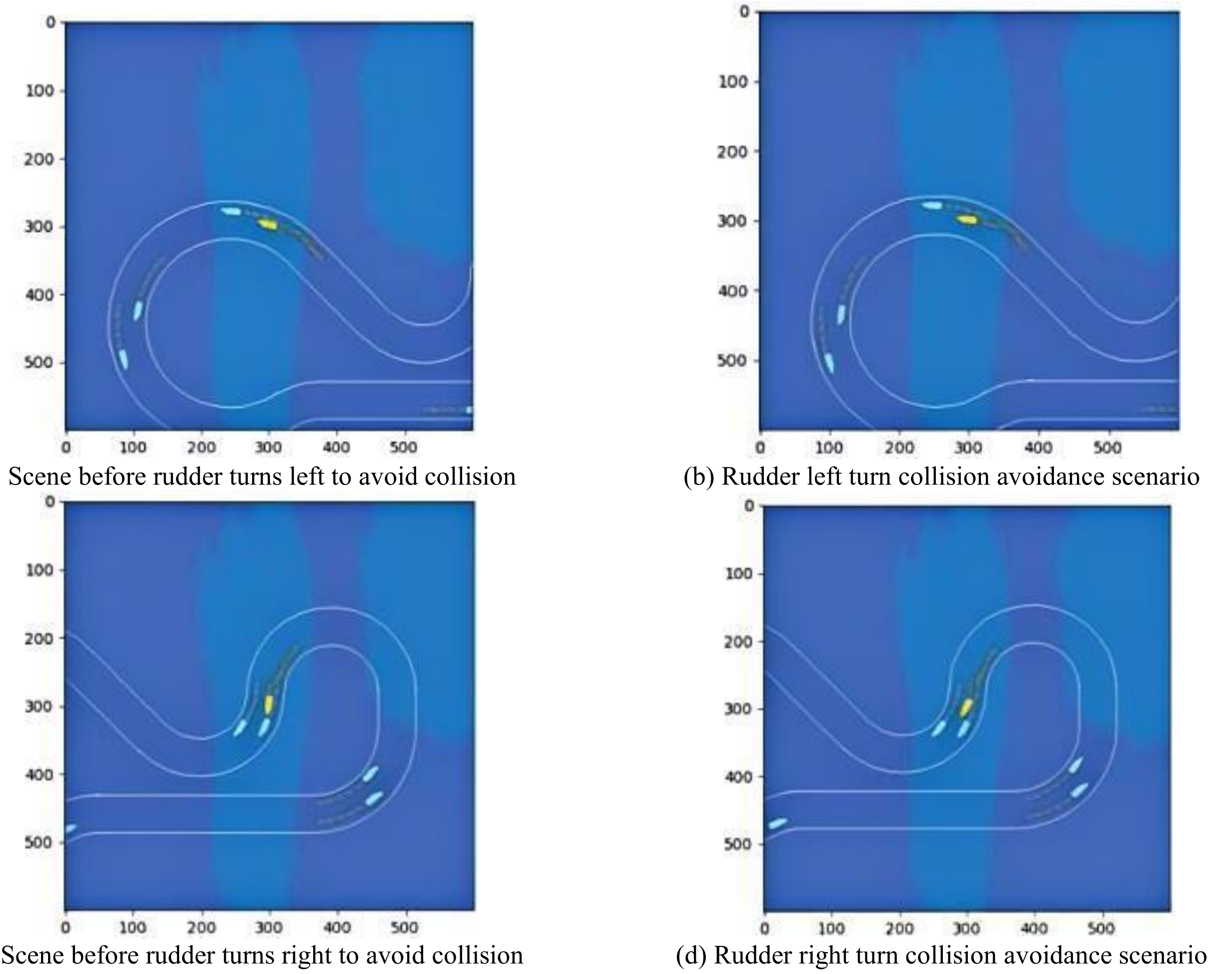
(a) The second frame of the scene before the steering collision

(b) The first frame of the scene before the steering collision

(c) The steering collision scene

Fig. 8 Collision diagram for steering maneuvers in a steering control simulation environment

Figure 9 illustrates the process of the ship agent avoiding collisions with other ships in the steering control intelligent collision avoidance simulation environment. In Figure 9a, the ship agent turns the rudder left. The ship risk calculation equation, as referenced in Section 2, is used to guide the collision avoidance operations. Using the improved PDPG algorithm, the ship agent performs collision avoidance based on the collision risk calculation. The agent turns the rudder left by $17.8°$ and accelerates at 1.1, m/s^2 to avoid other ships. During the training process of the improved algorithm, the reward value for turning left and accelerating in the current state is higher than the reward value for other actions. In Figure 9b, the collision avoidance decision made by the ship agent, based on the improved PDPG algorithm, successfully prevents a collision. In Figure 9c, the ship agent turns the rudder right by $16.4°$ and decelerates at -1.7, m/s^2 to avoid other ships, as shown in Figure 9d. Under the training of the improved PDPG algorithm, the actions taken by the ship agent maximize the cumulative reward in the current round, and the agent selects the action with the highest reward value in the current state. After continuous training with the improved PDPG algorithm, the ship agent progressively enhances its intelligent collision avoidance decisions during steering control. The ship can safely navigate complex waterways with changing steering conditions.



(a) Scene before rudder turns left to avoid collision

(b) Rudder left turn collision avoidance scenario

(c) Scene before rudder turns right to avoid collision

(d) Rudder right turn collision avoidance scenario

Fig. 9 Collision avoidance chart in a steering control simulation environment

The ship agent learns intelligent collision avoidance decisions using the improved PDPG algorithm. The simulation experiment results indicate that the higher the reward value obtained in each round of the continuous steering simulation, the more effective the learned steering control and collision avoidance decisions. However, different learning algorithms may lead to alternative decisions for the steering control collision avoidance problem. The ship agent explores various steering actions in a given state and aims to identify the action that yields the highest reward. Thus, the effectiveness of the steering control and intelligent collision avoidance decisions can be evaluated by comparing the average rewards obtained by the algorithm throughout the entire simulation environment.

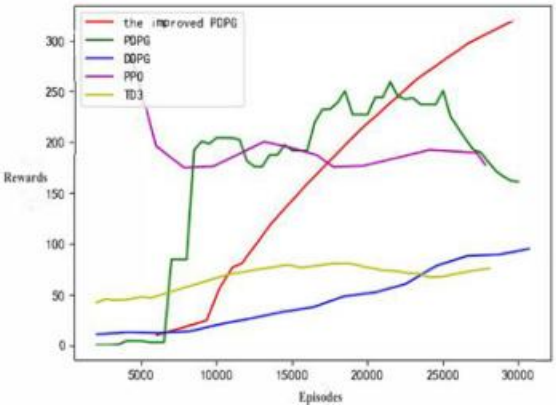


Fig. 10 Average rewards of comparative algorithms in continuous steering collision avoidance.

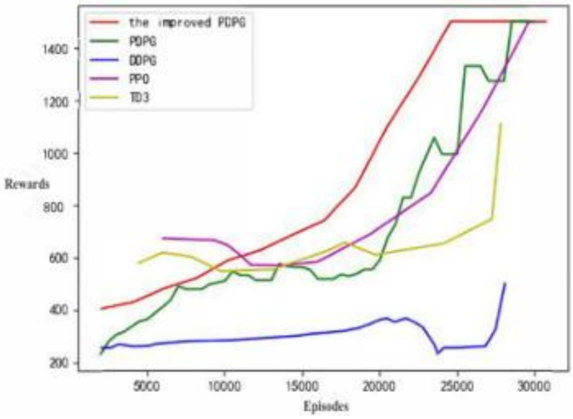


Fig. 11 Average steps of comparative algorithms in continuous steering collision avoidance.

The comparison algorithms were run independently 20 times, with each experiment consisting of 30,000 episodes in the simulation environment. Figure 10 shows the average reward graph for the comparison algorithms as the number of exploration steps increases. As seen in the figure, except for the TD3 algorithm, the reward values of the other comparison algorithms generally show an upward trend. However, as the number of rounds increases, algorithms such as PPO and PDPG exhibit a decreasing trend. In contrast, the reward value obtained by the improved PDPG algorithm increases steadily with the number of exploration steps, indicating that it has learned a more effective continuous steering control and intelligent collision avoidance strategy. Figure 11 presents the average number of simulation steps completed by the comparison algorithms throughout the simulation. All comparison algorithms exhibited robust exploration in the early stages, and as exploration continued, they all reached the maximum number of exploration steps. When considering both the average reward from Figure 10 and the average number of steps from Figure 11, the results indicate that the improved PDPG algorithm demonstrates superior performance in addressing the intelligent collision avoidance decision-making problem for ship steering control.

This paper presents a PDPG-based method for intelligent collision avoidance in continuous ship steering. It begins by analyzing the critical role of ship steering in navigation, highlighting that existing approaches often discretize actions or evaluate action quality, which can lead to local optimality in continuous steering-based collision avoidance. To overcome these limitations, the proposed method uses a neural network to learn continuous steering actions, with the network architecture adapted to identify optimal collision-avoidance strategies. In the constant action space, the balance between strategy exploration and exploitation is crucial to the quality of collision-avoidance action learning. The adaptive exploration rate and random strategy dynamically balance exploration and exploitation within the pathwise derivative policy gradient algorithm. The effectiveness of the improved method is validated through simulations of ship steering control.

To address the challenges posed by unknown disturbances, ship dynamics, and nonlinear control in the ship steering control model, the improved PDPG algorithm is proposed for intelligent collision avoidance decision-making in ship steering control. The main contributions of this paper are as follows: (1) Applying the path-derived policy gradient to resolve the limitation of traditional critics, which can evaluate the quality of a strategy but cannot provide the optimal strategy. (2) Adjusting the heading, rudder angle, and speed in the ship's intelligent collision avoidance decision using a reinforcement learning model, where the improved PDPG training network generates optimized control parameters. (3) Simulating the ship's steering control to produce intelligent collision avoidance decisions and comparing the improved model with other advanced models to validate the performance of the proposed algorithm.

5. Conclusion

This paper addresses the problem of intelligent collision avoidance decision-making in continuous steering control for ships by proposing a method based on the improved PDPG algorithm. The enhanced algorithm trains the ship agent to learn collision avoidance strategies for steering control in complex environments. Focusing on the ship MMG motion model, the study enhances the basic PDPG algorithm's strategy network structure by incorporating an adaptive exploration rate and dynamically balancing exploration and exploitation. This improved algorithm is well-suited for the nonlinear control of continuous states and actions.

To evaluate its performance, the algorithm is applied to simulations of ship steering control collision avoidance and compared with other algorithms, including DDPG, PPO, and TD3. The evaluation criteria include the collision and avoidance processes, average rewards from completed simulations, and the average number of steps. Experimental results demonstrate that the improved PDPG algorithm outperforms other reinforcement learning algorithms in intelligent collision avoidance decision-making for continuous steering control of ships. While the current simulation environment provides a suitable testbed for validating the proposed algorithm, it remains relatively simplified. In future research, we plan to extend this work in several directions:

- Complex multi-ship encounter scenarios: Introducing traffic-rich environments with varying interaction patterns to better reflect real-world maritime navigation challenges.
- Extreme meteorological conditions: Simulating adverse conditions such as strong winds, high waves, and reduced visibility to evaluate robustness under harsh environments.
- Cooperative decision-making: Incorporating COLREGs-compliant cooperative strategies for multiple autonomous vessels to enhance safety in shared waterways.

ACKNOWLEDGMENTS

This work was supported by the Suqian Science and Technology Project under Grant No.K202533 and the Information Engineering College Research Start-up Fee under Grant No.106-CK00042/167, the Jiangsu Social Science Foundation of China under Grant No.21YYB014, the National Natural Science Foundation of China under Grant No.62302330, the Natural Science Foundation of Jiangsu Province under Grant No.BK20230727.

REFERENCES

[1] Hou, L., Han, Y., Wang, R., 2025. Optimization design about end plates of ship rudder based on hydrodynamic numerical simulations. *Brodogradnja*, 76(4), 76404. https://doi.org/10.21278/brod76404

[2] Lisowski, J., Mohamed-Seghir, M., 2019. Comparison of computational intelligence methods based on fuzzy sets and game theory in the synthesis of safe ship control based on information from a radar ARPA system. *Remote Sensing*, 11(1), 82. https://doi.org/10.3390/rs11010082

[3] Perera, L. P., Carvalho, J. P., Guedes Soares, C., 2011. Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions. *Journal of Marine Science and Technology*, 16, 84-99. https://doi.org/10.1007/s00773-010-0106-x

[4] Yang, X., Han, Q., 2023. Improved reinforcement learning for collision-free local path planning of dynamic obstacle. *Ocean Engineering*, 283, 115040. https://doi.org/10.1016/j.oceaneng.2023.115040

[5] Zhang, H., Zhang, X., Bu, R., 2021. Active disturbance rejection control of ship course keeping based on nonlinear feedback and ZOH component. *Ocean Engineering*, 233, 109136. https://doi.org/10.1016/j.oceaneng.2021.109136

[6] Guan, W., Peng, H., Zhang, X., Sun, H., 2022. Ship steering adaptive CGS control based on EKF identification method. *Journal of Marine Science and Engineering*, 10(2), 294. https://doi.org/10.3390/jmse10020294

[7] Zhang, X., Xu, X., Li, J., Ma, F., Chen, Y., Xu, X., Shen, M., 2022. A novel switching control for ship course-keeping autopilot with steering machine bias failure and fault alarm. *Ocean Engineering*, 261, 112191. https://doi.org/10.1016/j.oceaneng.2022.112191

[8] Cao, Y., Zhang, J., Ma, A., Xu, H., Liu, J., 2026. Marine engine cylinder exhaust temperature prediction based on PSO-optimized CNN-LSTM-attention network. *Brodogradnja*, 77(1), 77101. https://doi.org/10.21278/brod77101

[9] Shin, G. H., Yang, H., 2025. Deep reinforcement learning for integrated vessel path planning with safe anchorage allocation. *Brodogradnja*, 76(3), 76305. https://doi.org/10.21278/brod76305

[10] Chen, Z., Qin, B., Sun, M., Sun, Q., 2020. Q-learning-based parameters adaptive algorithm for active disturbance rejection control and its application to ship course control. *Neurocomputing*, 408, 51-63. https://doi.org/10.1016/j.neucom.2019.10.060

[11] Sawada, R., Sato, K., Majima, T., 2021. Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces. *Journal of Marine Science and Technology*, 26(2), 509-524. https://doi.org/10.1007/s00773-020-00755-0

[12] Göksu, B., Yüksel, O., Şakar, C., 2023. Risk assessment of the Ship steering gear failures using fuzzy-Bayesian networks. *Ocean Engineering*, 274, 114064. https://doi.org/10.1016/j.oceaneng.2023.114064

[13] Zhang, X., Xu, X., Li, J., Ma, F., Zhang, Z., Brunauer, G., Steyskal, F., 2023. Fault estimation and H∞ fuzzy active fault-tolerant control design for ship steering autopilot subject to actuator and sensor faults. *IEEE Sensors Journal*, 23(22), 28110-28119. https://doi.org/10.1109/JSEN.2023.3321841

[14] Deraj, R., Kumar, R. S., Alam, M. S., Somayajula, A., 2023. Deep reinforcement learning based controller for ship navigation. *Ocean Engineering*, 273, 113937. https://doi.org/10.1016/j.oceaneng.2023.113937

[15] Qin, H., Tan, P., Chen, Z., Sun, M., Sun, Q., 2022. Deep reinforcement learning based active disturbance rejection control for ship course control. *Neurocomputing*, 484, 99-108. https://doi.org/10.1016/j.neucom.2021.06.096

[16] Waltz, M., Okhrin, O., 2023. Spatial–temporal recurrent reinforcement learning for autonomous ships. *Neural Networks*, 165, 634-653. https://doi.org/10.1016/j.neunet.2023.06.015

[17] Guan, W., Luo, W., Cui, Z., 2024. Intelligent decision-making system for multiple marine autonomous surface ships based on deep reinforcement learning. *Robotics and Autonomous Systems*, 172, 104587. https://doi.org/10.1016/j.robot.2023.104587

[18] Sivaraj, S., Dubey, A., Rajendran, S., 2023. On the performance of different deep reinforcement learning based controllers for the path-following of a ship. *Ocean Engineering*, 286, 115607. https://doi.org/10.1016/j.oceaneng.2023.115607

[19] Liu, J., Huang, L., Yu, D., Xu, L., He, Y., 2024. The control method for ship tracking when navigating through narrow and curved sections. *Applied Ocean Research*, 145, 103943. https://doi.org/10.1016/j.apor.2024.103943

[20] Neatby, H. C., Thornhill, E., 2024. Turning and stopping of a ship with twin Z-drive thrusters. *Ocean Engineering*, 293, 116641. https://doi.org/10.1016/j.oceaneng.2023.116641

[21] Chen, D., Dai, C., Wan, X., Mou, J., 2015. A research on AIS-based embedded system for ship collision avoidance. *In 2015 International Conference on Transportation Information and Safety* (ICTIS), June 25-28, Wuhan, China, 512-517. https://doi.org/10.1109/ICTIS.2015.7232141

[22] Li, L., Wu, D., Huang, Y., Yuan, Z. M., 2021. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Applied Ocean Research*, 113, 102759. https://doi.org/10.1016/j.apor.2021.102759

[23] Ortolani, F., Mauro, S., Dubbioso, G., 2015. Investigation of the radial bearing force developed during actual ship operations. Part 1: Straight ahead sailing and turning maneuvers. *Ocean Engineering*, 94, 67-87. https://doi.org/10.1016/j.oceaneng.2014.11.032

[24] Wang, C., Zhang, X., Gao, H., Bashir, M., Li, H., Yang, Z., 2024. COLERGs-constrained safe reinforcement learning for realising MASS's risk-informed collision avoidance decision making. *Knowledge-Based Systems*, 300, 112205. https://doi.org/10.1016/j.knosys.2024.112205

[25] Kwon, Y. J., 1981. The effect of weather, particularly short sea waves, on ship speed performance. *Doctoral dissertation*. Newcastle University, Newcastle, UK.

[26] Kepaptsoglou, K., Fountas, G., Karlaftis, M. G., 2015. Weather impact on containership routing in closed seas: A chance-constraint optimization approach. *Transportation Research Part C: Emerging Technologies*, 55, 139-155. https://doi.org/10.1016/j.trc.2015.01.027

[27] Liu, C., Li, T., Wu, W., Zheng, H., Li, J., Chu, X., 2024. Event-triggered predictive path following control of autonomous ships with an MMG model. *Ocean Engineering*, 314, 119582. https://doi.org/10.1016/j.oceaneng.2024.119582

[28] Leurent, E., 2018. An environment for autonomous driving decision-making.

[29] Li, B., Ouyang, Y., Li, L., Zhang, Y., 2022. Autonomous driving on curvy roads without reliance on frenet frame: A cartesian-based trajectory planning method. *IEEE Transactions on Intelligent Transportation Systems*, 23(9), 15729-15741. https://doi.org/10.1109/TITS.2022.3145389

[30] Hu, Y., Zhang, A., Tian, W., Zhang, J., Hou, Z., 2020. Multi-ship collision avoidance decision-making based on collision risk index. *Journal of Marine Science and Engineering*, 8(9), 640. https://doi.org/10.3390/jmse8090640

[31] Liu, J., Zhang, J., Yan, X., Soares, C. G., 2022. Multi-ship collision avoidance decision-making and coordination mechanism in Mixed Navigation Scenarios. *Ocean Engineering*, 257, 111666. https://doi.org/10.1016/j.oceaneng.2022.111666

[32] Betalo, M. L., Leng, S., Seid, A. M., Abishu, H. N., Erbad, A., Bai, X., 2025. Dynamic charging and path planning for uav-powered rechargeable wsns using multi-agent deep reinforcement learning. *IEEE Transactions on Automation Science and Engineering*. https://doi.org/10.1109/TASE.2025.3558945

[33] Betalo, M. L., Ullah, I., Tesema, F. B., Wu, Z., Li, J., Bai, X., 2025. Generative AI-Driven Multi-Agent DRL for Task Allocation in UAV-Assisted EMPD within 6G-Enabled SAGIN Networks. *IEEE Internet of Things Journal*, 12(17), 35890 – 35907. https://doi.org/10.1109/JIOT.2025.3579780

[34] Bhatnagar, S., Ghavamzadeh, M., Lee, M., Sutton, R. S., 2007. Incremental natural actor-critic algorithms. *Advances in neural information processing systems*, 20,

[35] Zanette, A., Wainwright, M. J., Brunskill, E., 2021. Provable benefits of actor-critic methods for offline reinforcement learning. *Proceedings of the 35th International Conference on Neural Information Processing Systems* (NIPS'21), December 6-14, Virtual, 34, 13626-13640.

[36] van Hasselt, H., Madjiheurem, S., Hessel, M., Silver, D., Barreto, A., Borsa, D., 2021. Expected eligibility traces. *In Proceedings of the AAAI conference on artificial intelligence*, February 2-9, Virtual, 35(11), 9997-10005. https://doi.org/10.1609/aaai.v35i11.17200

[37] Gronauer, S., Diepold, K. 2022. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2), 895-943. https://doi.org/10.1007/s10462-021-09996-w

[38] Abouelazm, A., Michel, J., Zöllner, J. M., 2024. A review of reward functions for reinforcement learning in the context of autonomous driving. *In 35th IEEE Intelligent Vehicles Symposium* (IEEE IV 2024), June 2-5, Jeju Shinhwa World, Jeju Island, Korea, 156-163. https://doi.org/10.1109/IV55156.2024.10588385

[39] Gao, Q., Li, S., Ji, Y., Liu, J., Song, Y., 2025. Scalable path planning algorithm for multi-unmanned surface vehicles based on multi-agent deep deterministic policy gradient. *Ocean Engineering*, 320, 120243. https://doi.org/10.1016/j.oceaneng.2024.120243

[40] Zheng, M., Zhang, J., Zhan, C., Ren, X., Lü, S., 2025. Proximal policy optimization with reward-based prioritization. *Expert Systems with Applications*, 127659. https://doi.org/10.1016/j.eswa.2025.127659

[41] Shu, M., Lü, S., Gong, X., An, D., Li, S., 2025. Episodic Memory-Double Actor–Critic Twin Delayed Deep Deterministic Policy Gradient. *Neural Networks*, 187, 107286. https://doi.org/10.1016/j.neunet.2025.107286